



Taller de Sistemas Operativos

Introducción

Agenda

- Introducción histórica (UNIX/Linux)
- Características Linux
- Proceso de inicialización del sistema (*Boot*)
- Proceso `init`

Introducción histórica

- UNIX

- Sistema operativo portable, multitarea y multiusuario
 - Originalmente desarrollado por personal de laboratorios Bell de AT&T, entre los que figuraban Ken Thompson y Dennis Ritchie



Introducción histórica

- UNIX: el sistema operativo original
 - Desde el punto de vista histórico, UNIX se refiere a la subfamilia de sistemas operativos que descienden de la primera implementación original de AT&T
 - El término descendencia ha de interpretarse como trabajos derivados que comparten propiedad intelectual con la implementación original
- UNIX: la familia
 - Desde el punto de vista técnico, UNIX se refiere a una familia de sistemas operativos que comparten criterios de diseño e interoperabilidad en común
 - Esta familia incluye más de 100 sistemas operativos desarrollados a lo largo de 20 años
 - Esta definición no implica necesariamente que dichos sistemas operativos compartan código o cualquier propiedad intelectual
- UNIX: la marca
 - Desde el punto de vista legal, Unix es una marca de mercado, propiedad de “The Open Group”, una organización de estandarización que permite el uso de dicha marca a cualquier sistema operativo que cumpla con sus estándares publicados, independientemente de que sea descendiente o clon del Unix original

Introducción histórica

- Portable
 - Instalable y compilable en diferentes plataformas (IA32, IA64, PowerPC, SPARC, Alpha, etc.)
- Multitarea
 - Gerencia tareas múltiples que ejecutan “simultáneamente”
 - Concepto de “procesos”
- Multiusuario
 - Una instalación en una máquina puede ser utilizada por múltiples usuarios simultáneamente
 - Acceso local o remoto
 - Privacidad (usuario/contraseña)

Introducción histórica

- El inicio del desarrollo de este sistema operativo (que se cree es uno de los más influyentes en la historia de la computación) fue muy particular, pues nadie habría predicho el éxito de UNIX después de su primera encarnación
- A finales de los años sesenta, el Instituto Tecnológico de Massachusetts, los laboratorios Bell de AT&T y General Electric trabajaban en un sistema operativo experimental llamado Multics (*Multiplexed Information and Computing System*), desarrollado para ejecutarse en un ordenador o computadora central (mainframe) modelo GE-645
- El objetivo del proyecto era desarrollar un sistema operativo interactivo que contase con muchas innovaciones, entre ellas mejoras en las políticas de seguridad. El proyecto, por cierto, consiguió dar a luz versiones para producción, pero las primeras versiones contaban con un pobre rendimiento
- Los laboratorios Bell de AT&T decidieron desvincularse y dedicar sus recursos a otros proyectos.
- Uno de los programadores del equipo de los laboratorios Bell, Ken Thompson, siguió trabajando para la computadora GE-645 y escribió un juego llamado Space Travel (Viaje espacial)
- Sin embargo, descubrió que el juego era lento en la máquina de General Electric

Introducción histórica

- De este modo, Thompson escribió nuevamente el programa, con ayuda de Dennis Ritchie, en lenguaje ensamblador, para que se ejecutase en un ordenador DEC PDP-7
- Esta experiencia, junto al trabajo que desarrolló para el proyecto *Multics*, condujo a Thompson a iniciar la creación de un nuevo sistema operativo para la DEC PDP-7
- Thompson y Ritchie lideraron un grupo de programadores, entre ellos a Rudd Canaday, en los laboratorios Bell, para desarrollar tanto el sistema de ficheros como el sistema operativo multitarea en sí. A lo anterior, agregaron un intérprete de órdenes (o intérprete de comandos) y un pequeño conjunto de programas
- El proyecto fue bautizado UNICS, como acrónimo de *Uniplexed Information and Computing System*, pues sólo prestaba servicios a dos usuarios (de acuerdo a Andrew Tanenbaum, era sólo un usuario)
- La autoría de esta sigla se le atribuye a Brian Kernighan, ya que era un experto de *Multics*. Dada la popularidad que tuvo un juego de palabras que consideraba a UNICS un sistema MULTICS castrado (pues eunuchs, en inglés, es un homónimo de UNICS), se cambió el nombre a UNIX, dando origen al legado que llega hasta nuestros días

Introducción histórica

- Hasta ese instante, no había existido apoyo económico por parte de los laboratorios Bell, pero eso cambió cuando el Grupo de Investigación en Ciencias de la Computación decidió utilizar UNIX en una máquina superior a la PDP-7
- Thompson y Ritchie lograron cumplir con la solicitud de agregar herramientas que permitieran el procesamiento de textos a UNIX en una máquina PDP-11/20, y como consecuencia de ello consiguieron el apoyo económico de los laboratorios Bell
- Fue así como por vez primera, en 1970, se habla oficialmente del sistema operativo UNIX ejecutado en una PDP-11/20. Se incluía en él un programa para dar formato a textos (*runoff*) y un editor de texto
- Tanto el sistema operativo como los programas fueron escritos en el lenguaje ensamblador de la PDP-11/20. Este "sistema de procesamiento de texto" inicial, compuesto tanto por el sistema operativo como de *runoff* y el editor de texto, fue utilizado en los laboratorios Bell para procesar las solicitudes de patentes que ellos recibían. Pronto, *runoff* evolucionó hasta convertirse en *troff*, el primer programa de edición electrónica que permitía realizar composición tipográfica
- El 3 de noviembre de 1971 se publicó The UNIX Programmer's Manual.

Introducción histórica

- En 1973 se tomó la decisión de escribir nuevamente UNIX, pero esta vez en el lenguaje de programación C. Este cambio significaba que UNIX podría ser fácilmente modificado para funcionar en otros ordenadores (de esta manera, se volvía portátil) y así otras variaciones podían ser desarrolladas por otros programadores. Ahora, el código era más conciso y compacto, lo que se tradujo en un aumento en la velocidad de desarrollo de UNIX. AT&T puso a UNIX a disposición de universidades y compañías, también al gobierno de los Estados Unidos, a través de licencias
- Una de estas licencias fue otorgada al Departamento de Computación de la Universidad de California, con sede en Berkeley. En 1975 esta institución desarrolló y publicó su propio sucedáneo de UNIX, conocida como Berkeley Software Distribution (BSD), que se convirtió en una fuerte competencia para la familia UNIX de AT&T
- Mientras tanto, AT&T creó una división comercial denominada Unix Systems Laboratories para la explotación comercial del sistema operativo. El desarrollo prosiguió, con la entrega de las versiones 4, 5 y 6 en el transcurso de 1975. Estas versiones incluían los pipes, lo que permitió dar al desarrollo una orientación modular respecto a la base del código, consiguiendo aumentar aún más la velocidad de desarrollo. Ya en 1978, cerca de 600 o más máquinas estaban ejecutándose con alguna de las distintas encarnaciones de UNIX

Introducción histórica

- La versión 7, la última versión del UNIX original con amplia distribución, entró en circulación en 1979. Las versiones 8, 9 y 10 se desarrollaron durante la década de 1980, pero su circulación se limitó a unas cuantas universidades, a pesar de que se publicaron los informes que describían el nuevo trabajo. Los resultados de esta investigación sirvieron de base para la creación de Plan 9, un nuevo sistema operativo portátil y distribuido, actualmente disponible en Laboratorios Bell
- AT&T entonces inició el desarrollo de UNIX System III, basado en la versión 7, como una variante de tinte comercial y así vendía el producto de manera directa. La primera versión se lanzó en 1982. A pesar de lo anterior, la empresa subsidiaria Western Electric seguía vendiendo versiones antiguas de Unix basadas en las distintas versiones hasta la séptima. Para finalizar con la confusión con todas las versiones divergentes, AT&T decidió combinar varias versiones desarrolladas en distintas universidades y empresas, dando origen al Unix System V Release 1. Esta versión presentó características tales como el editor Vi y la biblioteca curses, desarrolladas por Berkeley Software Distribution en la Universidad de California, Berkeley. También contaba con compatibilidad con las máquinas VAX de la compañía DEC

Introducción histórica

- En 1993, la compañía Novell adquirió la división Unix Systems Laboratories junto con su propiedad intelectual. Esto ocurrió en un momento delicado en el que Unix Systems Laboratories disputaba una demanda en los tribunales contra BSD por infracción de los derechos de copyright, revelación de secretos y violación de marca de mercado.
- BSD no solamente ganó el juicio sino que cambiaron tornas descubriendo que grandes porciones del código de BSD habían sido copiadas ilegalmente en UNIX System V. En realidad, la propiedad intelectual de Novell (recién adquirida de Unix Systems Laboratories) se reducía a unos pocos ficheros fuente. La correspondiente contra-demanda acabó en un acuerdo extrajudicial cuyos términos permanecen bajo secreto a petición de Novell
- En 1995, Novell vendió su división UNIX comercial (es decir, la antigua Unix Systems Laboratories) a Santa Cruz Operation (SCO) reservándose, aparentemente, algunos derechos de propiedad intelectual sobre el software
- SCO continúa la comercialización de System V en su producto UnixWare, que durante cierto tiempo pasó a denominarse OpenUnix, aunque ha retomado de nuevo el nombre de UnixWare

Introducción histórica

- Aproximadamente por esa misma fecha, un estudiante de ciencias de la computación llamado Linus Torvalds desarrolló un sistema operativo para ordenadores con arquitectura de procesador Intel x86 que mimetizaba muchas de las funcionalidades de UNIX
- Lo denominó Linux y lo lanzó en forma de código abierto en 1991
 - GNU General Public License (GPL)
- Cientos de miles de desarrolladores incorporaron otras funcionalidades en el núcleo de este sistema operativo derivando, después de unos años, en un sistema multiplataforma capaz de competir con las demás familias de UNIX
 - Septiembre de 1991 versión 0.01
 - Marzo de 1994 versión 1.0.0
 - Junio de 1996 versión 2.0.0
 - Enero de 1999 versión 2.2.0
 - Enero de 2001 versión 2.4.0
 - Diciembre de 2003 versión 2.6.0
 - Mayo de 2012 versión 3.4.0

Características Linux

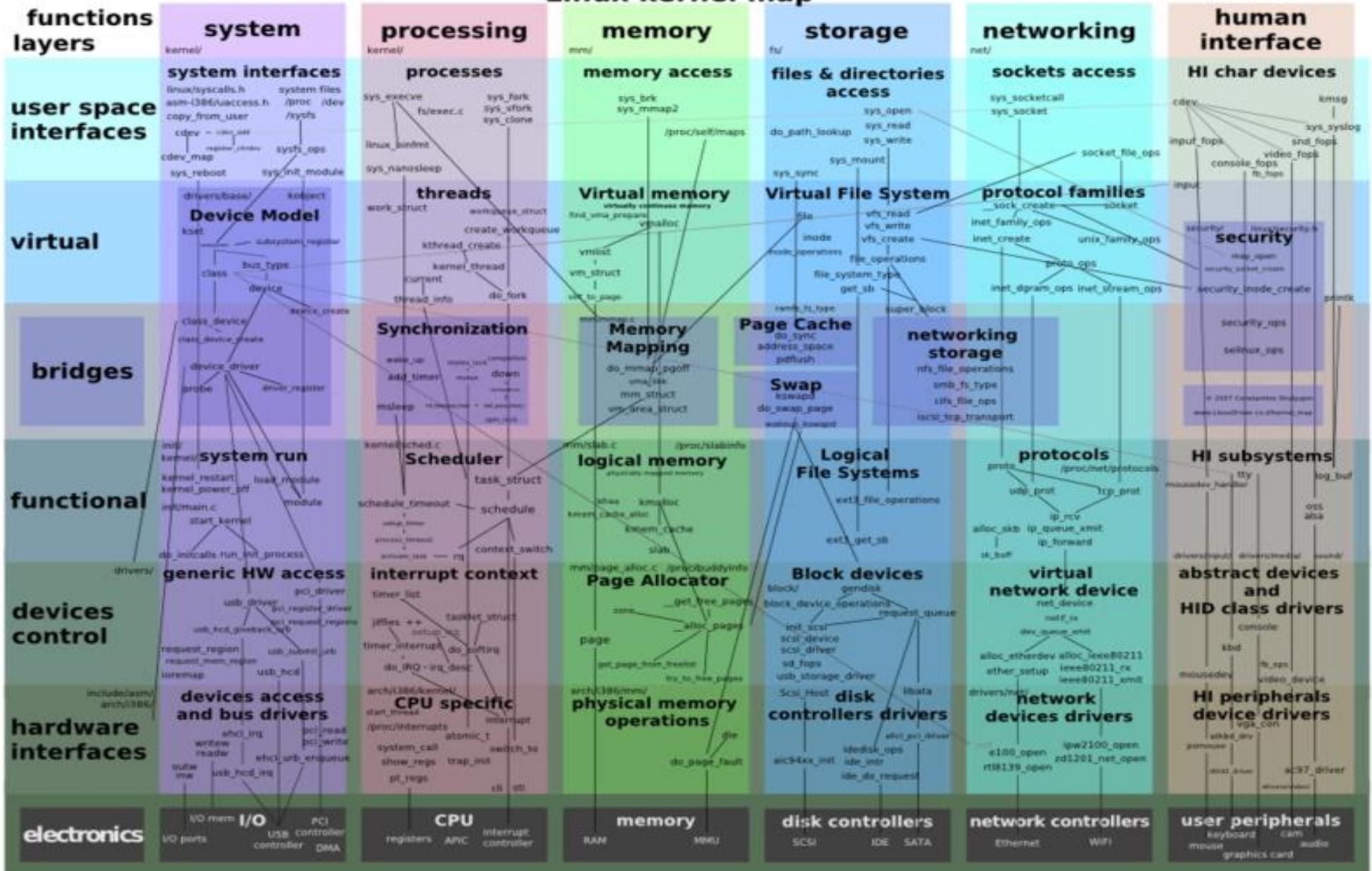
- Sistema multitarea
 - Varias tareas pueden ejecutarse en forma concurrente
- Multiprocesamiento simétrico
 - Reentrante
- Sistema multiusuario
 - Autenticación
 - Protección entre usuarios y programas del sistema
 - Mecanismos de conteo para limitar el uso a los usuarios
- Modelo Proceso/Núcleo
 - Servicios del sistema accedidos por los llamados al sistema

Características Linux

- Arquitectura monolítica
 - Utiliza módulos que agregan funcionalidad al núcleo
- Implementa el concepto de memoria virtual
- Expropia a nivel de Núcleo
- Prevención del *dead-locks* mediante el pedido de *locks* en un orden predefinido
- Brinda una interfaz común de manejadores de dispositivos
- Asociación en tiempo de ejecución (*address binding*)
- Librerías dinámicas
- Mapeo de memoria
- Multiplataforma: IA32, IA64, SPARC, Power, Alpha, etc.

Características Linux

Linux kernel map



Proceso de inicialización del sistema (Boot)

- Arquitectura IA32
 - Arquitectura CISC
 - Little-endian
 - Real-mode
 - Direccionamiento de 20bits (1Mb)
 - Acceso directo a rutinas de la BIOS (*Basic Input/Output System*) y componentes de hardware
 - Al iniciar el sistema se ejecuta el código a partir de una dirección de memoria física
 - Esa dirección tiene mapeada por hardware a la ROM
 - El programa que está en la ROM es la BIOS

BIOS

- Ejecuta ciertos test sobre el hardware disponible (llamado *Power-On-Self-Test*)
- Inicializa los dispositivos de hardware (garantiza que los dispositivos no hagan conflictos en líneas IRQ y puertos de E/S)
- Busca un sistema operativo para realizar la inicialización (*boot*) del sistema
- Una vez que un dispositivo es encontrado, carga el primer sector en memoria principal (0x00007c00), y ejecuta el código cargado

Inicialización a partir de un disco

- El primer sector de un disco duro, MBR (*Master Boot Record*), incluye una tabla y un pequeño programa
- La tabla contiene la información de particiones del disco
- El pequeño programa es por lo general un cargador (*boot loader*), que carga otro componente a partir de otra dirección y luego salta a esa dirección
- Por lo general los cargadores son de varias etapas (*multi-stage*)
- Esto es debido a una limitación de 512 *bytes* que presenta el *hardware*

Multi-stage boot loader y carga del núcleo

- El cargador realiza la siguiente tarea:
 - Invoca una rutina al BIOS para que despliegue un mensaje en pantalla
 - Invoca una rutina para que cargue los primeros 512 *bytes* de la imagen del núcleo a partir de la dirección 0x00090000 y consecutivamente carga el código de la rutina `setup` a partir de la dirección 0x00090200
 - Invoca al BIOS para que cargue el resto de la imagen del núcleo a partir de la dirección 0x00010000 (*zImage*) ó 0x00100000 (*bzImage*)
 - Salta a la dirección donde está el código de la función `setup` (0x00090200)

Inicialización del sistema operativo

- La función en lenguaje ensamblador `setup` realiza las siguientes tareas:
 - Reinicializa todo el hardware
 - Se invoca una rutina de la BIOS que construye una tabla en RAM que describe la disposición de la memoria principal (BIOS-e820)
 - Configura el retardo del teclado
 - Inicializa la placa de video
 - Inicializa el controlador de disco y determina los parámetros del disco
 - Busca por un bus microcanal (*Micro Channel Bus*)
 - Busca por un dispositivo de ratón (*mouse*)
 - Controla si hay soporte a nivel de BIOS para *Advanced Power Management*
 - Construye una tabla en memoria que describe los discos disponibles (EDD)
 - Reubica el código del núcleo a partir de la dirección 0x00001000 (caso *zImage*)

Inicialización del sistema operativo

- Se asegura que todas las interrupciones están deshabilitadas
- Incorpora una tabla de descriptores de interrupciones (IDT) y una tabla global de descriptores (GDT)
- Resetea la unidad de punto flotante
- Reprograma el controlador de interrupciones programable (PIC – *Programmable Interrupt Controller*)
- Cambia el modo del procesador a modo protegido
- Salta a la rutina en lenguaje ensamblador `startup_32`

Inicialización del sistema operativo

- La primer función `startup_32`
 - Inicializa los registros de segmentación y crea un stack provisional
 - Limpia el registro *EFLAGS*
 - Descomprime el kernel y lo mueve a la dirección física `0x00100000`
 - Salta a la dirección `0x00100000` (`startup_32`)

Inicialización del sistema operativo

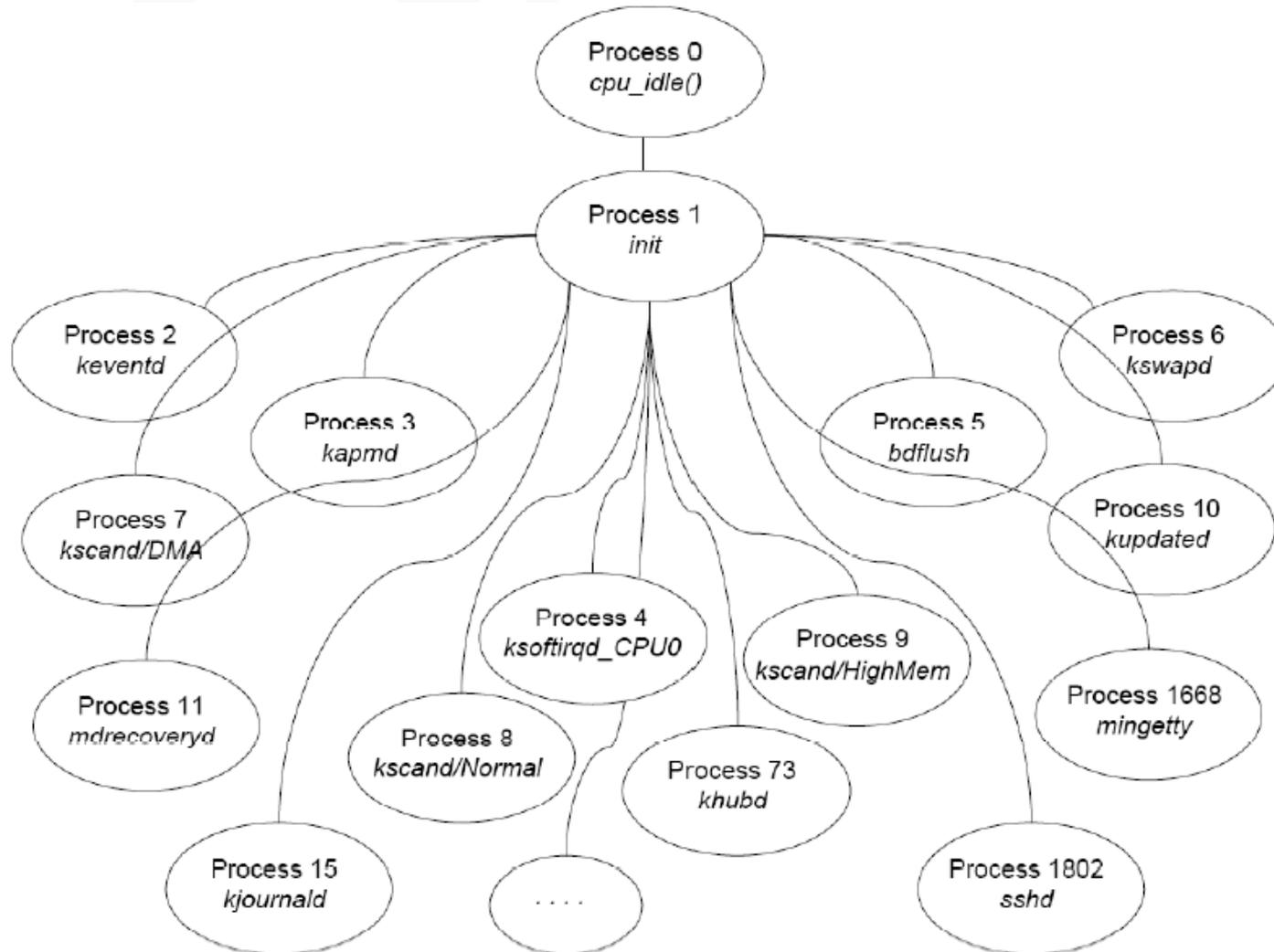
- La segunda función `startup_32` configura el ambiente para el primer proceso (proceso 0)
 - Inicializa los registros de segmentos con los valores finales
 - Carga la dirección del directorio global de páginas (*Page Global Directory*) en el registro `cr3` y se activa el sistema de paginación
 - Configura el *stack* para el proceso 0
 - Limpia todos los *bits* del registro de estado del procesador (*EFLAGS register*)
 - Inicializa las entradas de la IDT con punteros en nulo
 - Descomprime el kernel y lo posiciona a partir de `0x00100000`.
 - Salta a la dirección `0x00100000`

Inicialización del sistema operativo

- La función `start_kernel` (*init/main.c*) completa la inicialización de los componentes y estructuras del sistema
 - Inicializa el planificador, estructuras para representar la memoria, etc.
 - Configura la hora del sistema
 - Determina la velocidad del procesador
 - Habilita las interrupciones (`sti`)
 - Finalmente se invoca a la rutina `kernel_thread` que genera el *thread* para el proceso `init`
- El *thread* del proceso 0 invoca a la rutina `cpu_idle` (loop infinito)
- Se ejecuta la función `init` que crea otros *threads* del núcleo y finaliza invocando el proceso `/sbin/init`

Proceso init

- El proceso `/sbin/init` tiene como archivo de configuración el archivo `/etc/inittab`



init (Estilo System V)

```
# The default runlevel is defined here
id:3:initdefault:

# First script to be executed, if not booting in emergency (-b) mode
si::bootwait:/etc/init.d/boot

# /etc/init.d/rc takes care of runlevel handling
# runlevel 0 is System halt (Do not use this for initdefault!)
# runlevel 1 is Single user mode
# runlevel 2 is Local multiuser without remote network (e.g. NFS)
# runlevel 3 is Full multiuser with network
# runlevel 4 is Not used
# runlevel 5 is Full multiuser with network and xdm
# runlevel 6 is System reboot (Do not use this for initdefault!)
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
...
```

init (Estilo BSD)

```
# Default runlevel. (Do not set to 0 or 6)
id:3:initdefault:
# System initialization (runs when system boots).
si:S:sysinit:/etc/rc.d/rc.S
# Script to run when going single user (runlevel 1).
su:1S:wait:/etc/rc.d/rc.K
# Script to run when going multi user.
rc:2345:wait:/etc/rc.d/rc.M
# What to do at the "Three Finger Salute".
ca::ctrlaltdel:/sbin/shutdown -t5 -r now
# Runlevel 0 halts the system.
10:0:wait:/etc/rc.d/rc.0

# Runlevel 6 reboots the system.
16:6:wait:/etc/rc.d/rc.6
...
```